

基于 ARM 核的嵌入式 TCP/IP 协议栈简化实现

徐海军^{1,3}, 刘金刚^{1,2,3}, 王益华^{1,3}

(1. 计算机科学联合研究院, 北京 100037; 2. 中国科学院 计算技术研究所, 北京 100080; 3. 首都师范大学, 北京 100037)

摘要: 结合具体的工程实例,通过分析嵌入式 TCP/IP 协议栈的设计原理,研究并简化了嵌入式 TCP/IP 协议栈的通信模型、体系结构和实现方案。开发出基于 SAMSUNG S3C44B0X 微处理器和 RTL8019AS 以太网物理层接口的低功耗网络控制器,并通过 Socket 编程在嵌入式 μCLinux 系统上实现了网络通信,达到系统设计目标。

关键词: 嵌入式; TCP/IP; 套接字; ARM

中图法分类号: TP393. 04 文献标识码: A 文章编号: 1001-3695(2006)10-0251-03

Simplified Realization of Embedded TCP/IP Protocol Stack Based on ARM Core

XU Hai-jun^{1,3}, LIU Jin-gang^{1,2,3}, WANG Yi-hua^{1,3}

(1. United Research Institute of Computer Sciences, Beijing 100037, China; 2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China; 3. Capital Normal University, Beijing 100037, China)

Abstract: Joining an engineering example, by analysing Embedded TCP/IP protocol stack designing principle, research and simplify the protocol stack's communication model, architecture, scheme. And then, this paper gives a low consuming circuit of network communication based on the microprocessor of Samsung S3C44B0X, which takes RTL8019AS as physical layer interface. Finally successfully porting the embedded μCLinux on the microprocessor and realizes rightly network communicating by Socket programming, it achieve the design goal by testing.

Key words: Embedded System; TCP/IP; Socket; ARM

1 引言

随着计算机技术的发展,嵌入式系统已成为计算机领域的一个重要组成部分。嵌入式系统中的网络应用也越来越广泛,几乎所有设备均需要有 Internet 连接能力。支持嵌入式系统上网的标准大部分仍然是 TCP/IP 协议,包括 TCP, IP, ARP 等协议。但是传统的 TCP/IP 协议在实现实时性方面做得不够好,它把大量的精力花在保证数据传送的可靠性以及数据流量的控制上,而在实时性要求比较高的嵌入式领域中,传统的 TCP/IP 不能满足其实时要求。因此研究并改进嵌入式 TCP/IP 协议栈以满足嵌入式系统的高性能要求具有重要的现实意义。

2 嵌入式 TCP/IP 协议栈的基本特征

传统 TCP/IP 协议栈的实现过于复杂,需占用大量系统资源,而嵌入式应用系统的资源往往都非常有限。因此,需要将传统的 TCP/IP 协议栈在不违背协议标准的前提下加以改进,使其实现性得到提高,占用的存储空间尽可能少,以满足嵌入式应用的要求。通常为了解决存储能力不足的问题,采取在嵌入式系统的缓冲区内开辟较少字节的固定存储空间,而不是动态分配,若一旦出现大流量数据包时就会导致缓冲区溢出。因

此在设计嵌入式 TCP/IP 协议栈时要合理地控制中断处理程序大小,使运行时间尽可能地缩短。

3 嵌入式 TCP/IP 协议栈体系结构

按照 OSI 层次结构思想,在标准的 TCP/IP 协议栈中有很多处理协议,如 ARP, IP, ICMP, TCP, UDP 等协议。在本文设计过程中考虑到嵌入式设备资源有限和对网络要求程度不高,经设计简化后的嵌入式 TCP/IP 协议栈体系结构如图 1 所示。整个协议栈采用模块化设计思想,主要模块是 ARP 协议处理模块、IP 协议处理模块、TCP 协议处理模块等。同时为每个模块设计良好的通信接口,保证上层、下层协议的系统调用。

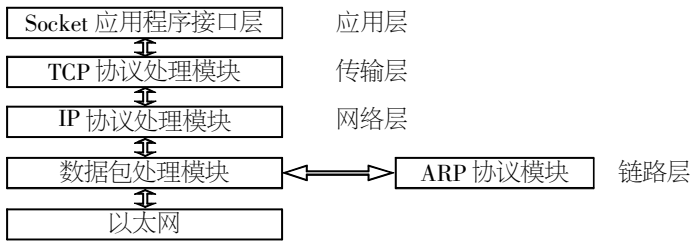


图 1 简化后的协议栈体系结构

由图 1 可知, TCP/IP 协议栈及以太网中数据传送的层次关系: 当在应用程序(一般应用有 HTTP, FTP 等协议)中将应用数据(包括用户数据和应用首部)向网络传送,它首先到达 TCP 层, TCP 协议根据应用层的要求在 TCP 首部填写好各个字段,如端口号、序号、标志等。重要的是填写数据校验和到校验和的字段,然后将包括 TCP 首部的段(数据包在 TCP 协议层称为段——Segment)向协议栈的下一层即 IP 层传送。IP 层则

与 TCP 层一样, 填写 IP 首部的各个字段, 如地址、协议类型等, 然后将在头部包括 IP 首部和 TCP 首部的整个数据报(数据包在 IP 协议层称为数据报——Datagram) 向下传送; 到了以太网驱动程序, 继续进行封装工作, 将以太网首部和以太网尾部添加到从 IP 层传下来的数据报上。

4 嵌入式 TCP/IP 协议处理流程

嵌入式 TCP/IP 协议接收数据包的实现过程就是在解析数据包。首先由以太网帧处理模块解析数据包, 根据其类型, 将去掉以太网帧首部的数据包分配到 IP 缓存或者 ARP 缓存; 接着, 由 IP 协议处理模块或 ARP 处理模块继续解析。ARP 协议根据数据包的类型, 或者更新 ARP 地址映射表或者发送 ARP 应答。IP 协议处理模块对数据包解析后, 将数据包交给 TCP 协议处理模块。

嵌入式 TCP/IP 协议发送数据包的实现过程可以认为是在封装数据包, 数据包经过某层协议的处理, 将上层协议传来的数据包封装成自己的格式。

按照改进后的嵌入式 TCP/IP 协议体系结构, 其协议的具体处理流程如图 2 所示。

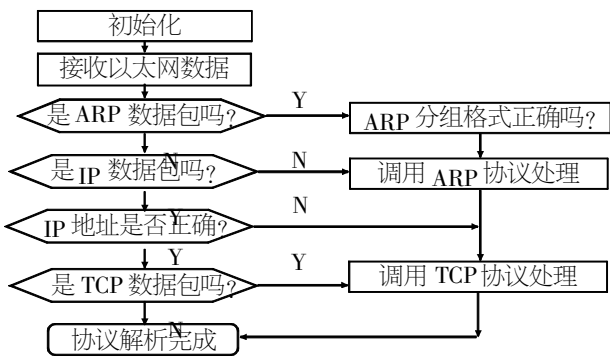


图 2 改进后的 TCP/IP 协议处理流程图

下面进一步对嵌入式协议栈中的 TCP 协议处理模型进行简化, 并通过 Socket 编程在 μ CLinux 系统上得以实现。

5 简化的嵌入式 TCP 协议设计实现

传输控制协议(Transmission Control Protocol, TCP) 是对 IP 协议进行功能扩展, 在发送端与接收端之间提供高可靠性的数据通信。TCP 协议是一个面向连接的通信协议, 在通信开始时建立连接; 在通信结束时切断该连接。在 TCP 协议中根据 IP 协议的载荷能力和物理网络最大传输单元 MTU 来决定数据段大小, 这些数据段称为 TCP 数据报报文。它由数据报头和数据两部分组成, 数据报头携带了该数据报所需的标志及控制信息, 包括 20Bytes 的固定部分和一个不固定长度的可选项部分。

在本次设计中, 根据系统网络数据量较小且 10Mbps 的以太网传输不会导致阻塞, 以太网上的主机有足够的处理能力处理数据报, 因此可以绑定超时与重传的时间为 5s; 所使用的 RTL8019AS 上有两个 1 500Bytes 的接收缓冲区, 对一般的嵌入式系统信息量较小, 可以固定接收窗口为 1 400Bytes; 采用一般的 TCP 服务就可以满足应用要求, 可以忽略紧急指针和选项及填充字段的值。因此对复杂的 TCP 协议进行了合理的改进, 改进后的格式如图 3 所示。

源端口(16 位)								目的端口(16 位)	
序列号(32 位)									
确认号(32 位)									
包头长度 (4 位)	保留 (6 位)	URG	ACK	PSH	PST	SYN	FIN	固定窗口	
数据(可变长)									

图 3 改进后的 TCP 协议数据报文格式

对于嵌入式 TCP/IP 协议栈的设计采用基于事件驱动的程序模型。当一个事件到达时(如一个新的连接请求或一个新的数据包到达一个已经建立的连接等), 应用程序就会被调用, 并由应用程序根据所发生的事件作出处理。此部分可以由具体的进程来实现。

(1) 建立连接: ①当客户机请求对端接入时, 随机地选送一个初始序号; ②服务器选送一个自己的初始序号, 作为对客户机送来序号的应答号返送给客户机; ③客户机向服务器再发出应答段(ACK), 作为握手信号来保证数据被可靠地接收, 而应答段本身不再需要应答, 避免应答陷入无穷的嵌套。

(2) 验证进程: 采取相应的措施消除传输中的错误, 以保证数据传输的可靠性, 如持续跟踪已发出数据段的应答是否返回来判断数据是否丢失; 利用序列号解决通信时重复、失序的问题; 利用校验和解决数据误码问题等。

(3) 流量控制进程: 设置一个缓冲区作为固定窗口, ACK 和窗口号指明在正确收到最后一个数据包之后, 还可接收的序列号范围, 由此对流量进行控制。

(4) 关闭连接: ①客户机向服务器发出关闭段, 此时客户机不再发出数据仅可接收数据; ②服务器向客户机发出关闭—应答段, 此时, 服务器还可以向客户机发送数据, 即接入处于“半关闭”状态; ③服务器向客户机发出关闭段, 服务器不能再发送数据; ④客户机为响应服务器的关闭, 向服务器发出关闭—应答段。改进后的基于事件驱动模型的 TCP 协议处理有限状态图如图 4 所示。

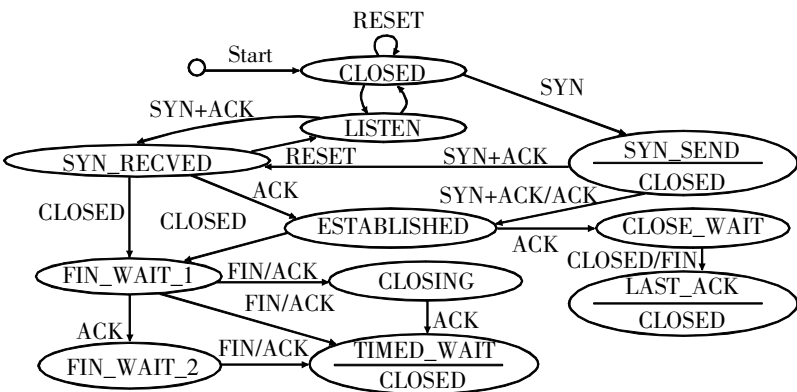


图 4 基于事件驱动模型的 TCP 协议处理有限状态图

SYN: 标志位用来建立连接, 让连接双方同步序列号。如果 SYN = 1 而 ACK = 0, 则表示该数据包为连接请求, 如果 SYN = 1 而 ACK = 1 则表示接收连接。

FIN: 表示发送端已经没有数据要求传输了, 希望释放连接。

RST: 用来复位一个连接。RST 标志置位的数据包称为复位包。一般情况下, 如果 TCP 收到的一个分段明显不属于该主机上的任何一个连接, 则向远端发送一个复位包。

ACK: 确认标志位。如果为 1, 表示包中的确认号是有效的; 否则, 包中的确认号无效。

接下来, 介绍如何在 μ CLinux 环境下通过 Socket 编程实现 TCP 协议。网络的 Socket 数据传输是一种特殊的 I/O, 也是一种文件描述符。具有一个类似文件的函数调用 Socket(), 该函数返回一个整型的 Socket 描述符, 随后的建立连接、数据传输等操作均是通过该 Socket() 函数实现的。在实现 TCP 协议过程中所使用的主要套接字系统调用、库函数和结构体如下所示:

```
#include < sys/types. h >
#include < sys/sockets. h >
#include < unistd. h >
int Socket ( int domain, int type, int protocol);
//利用 Socket 系统调用来打开一个套接字
int bind ( int s, struct sockaddr * my_addr, socklen_t addrlen);
//利用自己的主机指定所使用的 IP 地址和端口
int connect ( int s, const struct sockaddr * addr, socklen_t addrlen);
//当指定通信对方 IP 地址时, 采用 Connect 系统调用
int listen ( int s, int backlog);
//当服务器接收到 TCP 协议连接时, 执行一个 Listen 系统调用
int accept ( int s, struct sockaddr * addr, socklen_t * addlen);
//当服务器接收来自客户机建立的连接请求时, 执行 Accept 系统调用

int send ( int s, const void * msg, size_t len, int flags);
//在建立连接后, 利用 Send 系统调用完成报文的发送
int recv ( int s, const void * buf, size_t len, int flags);
//在建立连接后, 利用 Send 系统调用完成报文的发送
int close ( int s);
//在结束对套接字使用时, 利用 Close 系统调用来关闭套接字
```

根据嵌入式 TCP/IP 协议栈的设计模型, 其实现主要是实现基本协议如 TCP 协议、IP 协议、ARP 协议等。其余的像 PPP 那样的链路层协议需要在嵌入式 TCP/IP 协议栈下面的设备驱动程序中实现; 而像 HTTP, FTP 这样的应用层协议则要在嵌入式 TCP/IP 协议栈上面的应用层中实现。

6 基于 ARM7TDMI 核的网络控制器硬件平台设计

本文所设计的嵌入式网络控制器由三大部分组成, 分别是微处理器、存储器、RTL8019AS 全双工以太网控制器。SAMSUNG 公司的 S3C44B0X 是基于以太网应用系统高性价比的 16/32 位 RISC 微控制器, 内含一个由 ARM 公司设计的 16/32 位 ARM7TDMI RISC 处理器核, ARM7TDMI 为低功耗、高性能的 16/32 核, 最适合用于对价格及功耗敏感的应用场合。本文主要工作是笔者从事北京市高校研究生教育基金项目的研究子课题, 完成了控制器的硬件实现, 并在 S3C44B0X 微处理器上成功移植 μ CLinux 操作系统。除了实现网络通信的基本功能外, 还增加了一些应用层上的服务, 如 FTP, HTTP 协议。下面是基于 ARM 核的嵌入式网络控制器的硬件结构设计方案如图 5 所示。

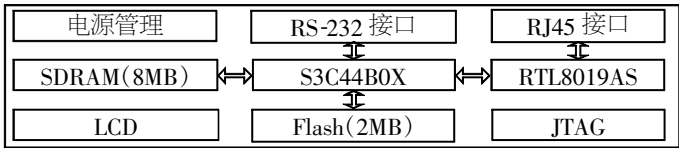


图 5 基于 ARM 核的嵌入式网络控制器的硬件结构

7 结束语

嵌入式 Internet 技术将随着网络和硬件技术的发展而不断提高, 网络化的嵌入式产品的流行成为一种不可抵挡的潮流。通过研究简化了基于 ARM 核的嵌入式 TCP/IP 协议栈, 实验

将本网络控制器的 IP 设为 192. 168. 0. 62, 连接好网线(一端插在控制器 RJ45 接口上, 另外一端插在 PC 机的网卡上), 在 PC 机的命令输入窗口中运行 Ping 程序, 输入 Ping 192. 168. 0. 62 -t, 回车结果一切正常(显示: Relay from 192. 168. 0. 62: bytes = 32 time < 1ms TTL = 20)。表明该系统能够使数据流按照改进后的网络协议模型进行处理, 并能正确地实现数据包的以太网传输。本文为嵌入式设备接入 Internet 提供了一种低成本、高可靠性、灵活性的解决方案。此外用户还可以根据具体设计需要, 开发出适合自己的网络微控制器。

参考文献:

[1] ung-Woo Tak, Jong-Moon Son, Tae-Keum Kim. Experience with TCP/IP Networking Protocol S/W over Embedded OS for Network Appliance[C] . ICPP Workshops, 1999. 556-561.

[2] 郝京宇, 黄建华, 贾风根. 协议栈的实现方法[J] . 计算机工程, 2004, 30(14) : 93-94.

[3] [美] W Richard Stevens. TCP/IP 详解(卷一) : 协议[M] . 范建华, 等. 北京: 机械工业出版社, 2000.

[4] [美] Gary R, Wrigh W, Richard Stevens. TCP/IP 详解(卷二) : 实现[M] . 陆学莹. 北京: 机械工业出版社, 2000.

[5] 李立清, 路海. 基于嵌入式系统的 TCP/IP 协议栈的实现[J] . 计算机工程, 2004, 30(19) : 83-84.

[6] (日) 村山公保. TCP/IP 网络实验程序篇[M] . 北京: 科学出版社, 2003.

[7] 郝身刚, 刘金江. 嵌入式系统中 TCP/IP 协议栈的研究[J] . 南阳师范学院学报(自然科学版), 2004, 23(3) : 78-80.

[8] 郝洁, 王慕坤, 付金生. 嵌入式 TCP/IP 技术研究及应用[J] . 哈尔滨理工大学学报, 2004, 9(2) : 100-102.

作者简介:

徐海军(1980-), 男, 四川仪陇人, 硕士研究生, 主要研究方向为智能接口技术; 刘金刚(1963-), 男, 辽宁人, 研究员, 博导, 主要研究方向为智能接口技术。

下期要目

- Java 多线程测试策略及测试方法探讨
- 语义块的多元逻辑组合构成研究
- 集装箱海运收益管理超订模型研究
- 基于逆向工程的领域本体开发方法
- 一种基于上三角项集矩阵的频繁项集挖掘算法
- 需求驱动的语义模型构建方法研究
- 基于数据挖掘的搜索引擎技术
- 软件连接件风格的研究
- 证据理论悖论消除的一种新方法
- 普适计算环境下室内定位模型的研究
- 一种基于条纹扫描的结构光三维信息获取方法
- 基于网格模型的运动估计技术研究
- 带功率捕获的随机接入算法性能分析
- 数域筛法中线性代数的硬件设计
- 变尺度动态网格法在数据分发管理中的研究
- 基于中间件的实时 VoD 系统研究
- 面向软件大规模定制的构件化领域框架研究
- 基于知识的数控弯管干涉回避算法
- 一种快速灾难恢复方法